INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATIONS AND ROBOTICS
ISSN 2320-7345

AN IMPRESSION ON PERFORMANCE METRICS FOR SCHEDULING PROBLEM IN GRID COMPUTING ENVIRONMENT

D. Thilagavathi[1], Dr. Antony Selvadoss Thanamani[2]

[1]*PhD Research Scholar, Research Department of Computer Science,*
NGM College, Pollachi – 642001.
[2]*Professor and Head, Research Department of Computer Science,*
NGM College, Pollachi – 642001.

Abstract

Grid Computing is a parallel and distributed computing. It helps to solve many large-scale scientific problems by effectively using the heterogeneous and geographically distributed resources. The job scheduling problem is a prominent research area in grid computing. Various scheduling algorithms are proposed for solving scheduling problem which is a NP-Hard Problem. The performance of scheduling algorithms is measured using various performance metrics. The choice of performance metrics for the researchers is really a challenging task. Several key performance metrics are needed to show the beneficiary factors of the behavior of various scheduling schemas. In this paper, we discuss a range of different metrics used by researchers for proving the competence of their algorithm.

Keywords: Grid Computing, Job scheduling, NP-Hard, Performance metrics.

Introduction

In grid environment, resources are heterogeneous, which are owned by multiple organizations and individuals. Each one follows different policies and has different security requirements. They are geographically separated, which are connected by heterogeneous multilevel networks. Resources and Environments are unreliable. Grid concept is meant as resource sharing and problem solving in dynamic, multi-institutional virtual organization. We need grid computing as it exploits the underutilized resources, increases throughput and reduce the idle time of resources. Actually in grid environment many systems are virtualized into one and given to one application.

Grid scheduling is a major area in grid computing. Various Scheduling Algorithms [10] are proposed for the grid scheduling problem. They are evaluated using a variety of performance metrics. Performance metrics helped the researchers to prove the competence of their algorithm.

The remainder of the paper is structured as follows. Section 2, Grid Scheduling Problem is described; Section 3 presents the Performance Metrics related to job scheduling which details the performance metrics used by different researchers to evaluate their scheduling algorithm. Importance of the performance metrics is discussed in Section 4. Finally, Section 5 concludes the paper.

1. Grid Scheduling Problem

The scheduling problem is a NP-Hard problem. It is not possible to find the optimal solutions in polynomial time. The grid scheduling is characterized by the heterogeneity of jobs, multi-objectiveness, dynamic nature of grids (i.e., new job arrives, new resources arrive or leave dynamically), heterogeneity of resources, heterogeneity of networks, etc.

The schedulers in grid systems called resource brokers aims on enhancing the performance of a specific application in such a way that its client requirements are met. The scheduling algorithm selects appropriate resource from n number of suitable resources. The scheduling decision should be good as well as timely decision is needed. If scheduling is not good then we want to reschedule specifically to migrate all data again. As rescheduling is costly, we need to find an optimal scheduling plan while scheduling the tasks for the first time itself.

The general formulation of any scheduling problem proposes to find an assignment of tasks to machines in order to satisfy some objectives. Given that using the computational resources does not come for free, most scheduling problems mainly concerns about time, trying to minimize the total time spent to execute all tasks. While several characteristics or properties can be assigned to each task (i.e. CPU and memory requirements, priority, execution time deadline, etc.), the simplest model used for scheduling, and also the most common one, considers only the execution time of every single task in each computation resource. So, the most usual metric to minimize in this model is the *makespan*, defined as the time spent from the instant when the first task begins to execute to the moment when the execution of the last task is completed. However, many other metrics have been considered, such as the economic cost of executing an application, and the quality of service, especially in grids infrastructures. Beyond all those single-objective approaches, scheduling is an obvious multiobjective problem when considering several task properties or several metrics in conflict with each other.

2. Performance Metrics

Individual users and system administrators often have different (and possibly conflicting) demands; no single measure can comprehensively capture overall grid performance. So, Grid system performance has been measured by various performance metrics. Table 1 shows the range of performance metrics used by different researchers for comprehensively evaluate their proposed scheduling algorithm. The various performance metrics are as follows:

- *Average response time* - The response time of a task is the time period between the task's arrival and its completion and the average response time is the average value of all tasks' response time. If N denote the total number of simulated jobs, C_i denote the completion time for a single job J_i, the arrival as a_i , b_i denote the start time for a single job J_i then ART is defined as:

$$\frac{\sum_{t=1}^{N}(c_i - a_i)}{N}$$

- *Average wait time (AWT)* - AWT is defined as:

$$\frac{\sum_{t=1}^{N}(c_i - b_i)}{N}$$

- *Site utilization* - The percentage of total task running time out of total available time of a given site
- *Makespan(C_{max})* - This is the time interval between the time at which the schedule begins and the time at which the schedule ends. Thus, the makespan of a schedule is equal to max[C_i], where i = 1, …, m.
- *Slowdown ratio* - The slowdown of a task is the ratio of the task's response time to its service time and the slowdown ratio is the average value of all tasks' slowdowns
- *Number of risk-taking job* N_{risk} - When sites provided security level cannot satisfy the jobs security demand, N_{risk} counts the number of jobs that are running on such kinds of sites.
- *Number of failed jobs* N_{fail} - Job execution may fail owing to insecure resource sites applied. N_{fail} counts the number of failed and rescheduled jobs. N_{fail} is bounded above by N_{risk}.

- *Flowtime* - The sum of finalization times of all the jobs
- *Tardiness* (T_i) - The tardiness T_i of a job J_i is then on-negative amount of time by which the completion time exceeds the due date d_i, Ti = max[0,(C_i - d_i)]. The differences between the completion time and due date for each job.
- *Weighted Grid Utilization (UTIL)* – It measures the overall ratio between consumed and available computational resources across a grid.

$$\text{UTIL} = \frac{\sum_j (ET_j - ST_j) \text{ X } CPU_j \text{ X } Clock_j}{(ET_{last} - QT_{first}) \text{ X } \sum_m CPU_m \text{ X } Clock_m}$$

where QT_j, ST_j and ET_j are the times when job j is queued to the grid, when it starts execution, and when it ends. (ET_{last} – QT_{first}) is the duration of the entire simulation, CPU_j and $Clock_j$ are the number of processors used by job j and their clock speed; and CPU_m and $Clock_m$ are the number of processors in machine m and their clock speed.

- *Fraction of Jobs Migrated* (FOJM) – It allows us to determine if there is any relationship between the number of jobs transferred and the performance of the scheduling algorithms.

$$\text{FOJ} = \frac{\text{Number of Jobs Transferred}}{\text{Total number of Jobs}}$$

- *Fraction of Data Volume Migrated* (FDVM) – It helps to determine the amount of data transferred by a scheduling algorithm affects its performance.

$$\text{FDVM} = \frac{\sum_k (\text{Input Size}_k + \text{Output Size}_k)}{\sum_k (\text{Input Size}_j + \text{Output Size}_j)}$$

- *Data Migration Overhead (DMOH)* - It is the fraction of job response time that is spent moving data, across all jobs.

$$\text{DMOH} = \frac{\text{Total Data Migration Time}}{\sum_j (ET_j - QT_j)}$$

- *Degree of security deficiency* - a weighted sum of *q* discrepancy values between security levels requested by a task and the security levels offered by a site
- *Speedup* - Is the third metric employed for the evaluation purpose of the algorithms which is incurred by dividing the sequential execution times of the graphs by their parallel execution times.
- *Efficiency* – It is the speedup and makespan ratio of the graph.
- *Turnaround time* - It is an important parameter for determining the performance of different fault tolerance techniques. It is the only parameter users pay attention for. It can be defined as the interval of time elapsed from submission of a job to the time of its completion.
- *Throughput* - Throughput is used to measure the ability of the grid to accommodate jobs. It is defined as:

$$Throughput(n) = \frac{n}{T_n}$$

where n is the total number of jobs submitted and T_n is the total amount of time necessary to complete n jobs.

- *Grid load*: It represents the amount of extra computations encountered by the grid to alleviate the effect of resources failures.
- *Fail tendency*: It is the percentage of the tendency of grid resources to fail and is defined as:

$$Fail\ Tendency = \frac{\sum_{j=1}^{m} P_{fj}}{m} \% \times 100\%$$

where *m* is the total number of grid resources and P_{fj} is the failure rate of resource *j*. Through this metric, we can expect the faulty behavior of the system.

- Success Rate: It tells how many jobs run successfully in a particular time.

Table 1: Various Performance metrics used in Grid Scheduling Environment

S.No	Performance Metric	Reference	Description
1	Average response time	[1], [3], [8]	It represents how fast a user receives a response from the system after the job is submitted.
2	Average wait time	[3], [8]	It must be minimized to increase the user's quality of service requirements.
3	Site utilization	[1], [4]	Algorithm must try to maximize this measure to increase the throughput of a grid.
4	Makespan	[1], [6], [4]	To minimize this objective, large tasks are mapped to the quicker machines. It measures the throughput of the system.
5	Slowdown ratio	[1], [4]	These three measures help to determine the Job failure rate.
6	Number of risk-taking jobs	[1]	
7	Number of failed jobs	[1]	
8	Flowtime	[6]	It is minimized by minimizing average task completion time, which is done by mapping small tasks to the fastest machines. It measures systems QOS.
9	Tardiness	[9]	Overall tardiness is reduced using slack rule: slack=Due date - Process time of a Job.
10	Weighted Grid Utilization	[3]	It helps in measuring the utilization of the available computational resources at the site.
11	Fraction of Jobs Migrated	[3]	FOJM captures only the number of migrated tasks but does not consider their data requirements.
12	Fraction of Data Volume Migrated (FDVM)	[3]	FDVM considers the migration of data.
13	Data Migration Overhead	[3]	It basically captures the overhead of grid scheduling.
14	Degree of security deficiency (DSD)	[4]	For a small task, a small DSD value means a high satisfaction degree. Zero DSD values say that task's security requirement is perfectly met.

15	Turnaround time	[8]	This measure is used to determine the fault tolerance of a grid environment
16	Throughput	[7]	It is accomplished by scheduling the maximum number of tasks to the grid in a certain period of time. Minimizing makespan tends to maximize throughput and balance load.

3. Importance of Performance Metrics

- Performance metrics helps in measuring significant work.
- Unmeasured work can be minimized or eliminated.
- Desired outcomes are necessary for work evaluation.
- It helps in timely corrective action.
- Work that is not measured or assessed cannot be managed because there is no objective information to determine its value.

4. Conclusion

In this paper, we have studied the grid scheduling problem. We identified and reviewed various performance metrics used in grid computing. Uses and importance of performance metrics are discussed. Analyzing the choice of performance metrics for a researcher is really a tedious job. We believe that job scheduling problem in Grid Computing, an area full of challenges and of principal importance, is still in its childhood now, and many research problems are yet to be identified. We predict several possible directions for future research on this area.

REFERENCES

1) Song, Shanshan, Yu-Kwong Kwok, and Kai Hwang, 2005, "Security-driven heuristics and a fast genetic algorithm for trusted grid job scheduling." *Parallel and Distributed Processing Symposium, 2005. Proceedings. 19th IEEE International*. IEEE.

2) Li, Keqin., 2004, "Experimental performance evaluation of job scheduling and processor afllocation algorithms for grid computing on metacomputers." *Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International*. IEEE.

3) Shan, Hongzhang, et al., 2004, "Scheduling in heterogeneous grid environments: The effects of data migration." *International Conference on Advanced Computing and Communication, Gujarat, India*.

4) Xie, Tao, and Xiao Qin, 2007, "Performance evaluation of a new scheduling algorithm for distributed systems with security heterogeneity." *Journal of Parallel and Distributed Computing* 67.10: 1067-1081.

5) Ijaz, Samia, et al., 2013, "Efficient scheduling strategy for task graphs in heterogeneous computing environment." *Int. Arab J. Inf. Technol.* 10.5: 486-492.

6) Xhafa, Fatos, et al., 2012, "A tabu search algorithm for scheduling independent jobs in computational grids." *Computing and informatics* 28.2: 237-250.

7) Amoon, M., 2011, "A Development Of Fault-Tolerant And Scheduling System For Grid Computing." *Computer Science & Telecommunications*32.3.

8) Selvi, Thamarai, 2010, "Nonpreemptive Priority (NPRP) based Job Scheduling model for virtualized grid environment." *2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*. Vol. 4.

9) Lorpunmanee, Siriluck, et al., 2007, "An ant colony optimization for dynamic job scheduling in grid environment." *International Journal of Computer and Information Science and Engineering* 1.4: 207-214.

10) Thilagavathi, D. and A. S. Thanamani, 2012, *A Survey on Dynamic Job Scheduling in Grid Environment Based on Heuristic Algorithms,* International Journal of Computer Trends and Technology Vol. 3, Issue 4, pp. 531-536.

A Brief Author Biography

Ms. D. Thilagavathi received her MCA degree from Bharathidasan University in 2001 and completed her M.Phil. degree in Computer Science from Bharathiar University in 2005. She is currently pursuing her Ph.D. at the Research Department of Computer Science, NGM College, Pollachi, under Bharathiar University, Coimbatore. Her research interests include Object Oriented Analysis and Design, Grid Computing and Cloud Computing. She has 13 years of teaching experience. She has published many papers in National conferences and International journals. She is presently working as an Assistant Professor and Head, Department of Computer Technology, NGM College, Pollachi.

Dr. Antony Selvadoss Thanamani is presently working as Professor and Head, Research Department of Computer Science, NGM College, Pollachi, Coimbatore, India. He has published more than 100 papers in international/national journals and conferences. He has authored many books on recent trends in Information Technology. His areas of interest include E-Learning, Knowledge Management, Data Mining, Networking, Parallel and Distributed Computing. He has to his credit 25 years of teaching and research experience. He is a senior member of International Association of Computer Science and Information Technology, Singapore and Active member of Computer Science Society of India, Computer Science Teachers Association, NewYork.